

CoAP and IoT

<https://jaime.win/lecture/coap>

Jaime Jiménez

jaime.jimenez@ericsson.com

[@jaim](#)

3rd May 2019

Table of Contents

1. IETF Standards
2. CoAP
3. Implementations and examples

Internet Engineering Task Force (IETF)



The IETF mission

- IETF's role: Specify the underlying, fundamental Internet technologies
- “Permissionless innovation” - others can build on top - unlike “App Stores” or Telco services.
- RFC3935: “The goal of the IETF is to make the Internet work better.”
- Some well known achievements ...
 - [RFC791](#) The Internet Protocol.
 - [RFC792](#) The Internet Control Message Protocol.
 - [RFC821](#) The Simple Mail Transfer Protocol.
 - [RFC768](#) User Datagram Protocol.
 - [RFC959](#) The File Transfer Protocol.
 - [RFC793](#) The Transmission Control Protocol.
 - [RFC854](#) Telnet Specification.
 - [RFC1119](#) Network Time Protocol.
 - [RFC1157](#) A Simple Network Management Protocol.
 - [RFC1035](#) Domain names - implementation and specification.
 - [RFC1945](#) Hypertext Transfer Protocol.
 - [RFC2131](#) Dynamic Host Configuration Protocol.
 - [RFC3261](#) The Session Initiation Protocol.
 - [RFC6455](#) The WebSocket Protocol.
 - [RFC5321](#) Simple Mail Transfer Protocol.
 - [RFC7540](#) Hypertext Transfer Protocol Version 2.
 - [RFC6749](#) The OAuth 2.0 Authorization Framework.
 - [RFC4271](#) The Border Gateway Protocol.
 - [RFC4287](#) The Atom Syndication Format.
 - [RFC4251](#) The Secure Shell (SSH) Protocol Architecture.
 - [RFC8200](#) Internet Protocol, Version 6 (IPv6) Sepcification.



"Rough consensus and running code"

Ways of working: RD Example

Working on the draft

<https://github.com/core-wg/resource-directory>

Working on the code

<https://github.com/thomas-fossati/gordo>

Other examples

<https://github.com/Ell-i/coap-rd>

<https://github.com/nning/core-rd>

The collage illustrates various GitHub workflows:

- Top Left:** A GitHub pull request page for the repository `core-wg/resource-directory`. It shows a list of pull requests with filters for 'is:pr is:closed' and 'Clear current search query, filters, and sorts'. The list includes pull requests for 'Explicitly state that a Registree-EP may keep multiple registra...', 'Reactions to 7ac551239', 'base text a bit stronger, registree -> registrant', 'security policies replaces security example, comments jaime', 'appendix B and D, and editorials', 'Summarize URI and relative-reference definitions', 'Talk of "relative references" and "URIs" rather than o', and 'christian's suggestions'.
- Top Right:** A GitHub pull request page for the repository `core-wg/resource-directory`, specifically for the pull request 'Reactions to 7ac551239 #161'. It shows the pull request details, including the commit hash `chrysn merged 7 commits into master from chrysn:261809` on 27 Sep 2018. The diff shows changes to `draft-ietf-core-resource-directory.md`, with a diff of `+62 -56` lines.
- Bottom Left:** A GitHub pull request page for the repository `core-wg/resource-directory`, showing a list of pull requests. The list includes pull requests for 'Explicitly state that a Registree-EP may keep multiple registra...', 'Reactions to 7ac551239', 'base text a bit stronger, registree -> registrant', 'security policies replaces security example, comments jaime', 'appendix B and D, and editorials', 'Summarize URI and relative-reference definitions', 'Talk of "relative references" and "URIs" rather than o', and 'christian's suggestions'.
- Bottom Right:** A screenshot of a code editor showing a Go program for a CoAP server. The code is in `main.go` and includes a `main` function that registers a CoAP endpoint and handles requests. The terminal window at the bottom shows the installation of dependencies: `Installing github.com/mdempsky/gocode SUCCEEDED`, `Installing github.com/uudashr/gopkgs/cmd/gopkgs SUCCEEDED`, `Installing github.com/ramya-rao-a/go-outline SUCCEEDED`, and `Installing github.com/acroca/go-symbols SUCCEEDED`.

Constrained Application Protocol (CoAP)



IETF: dozen+ years of IoT standards



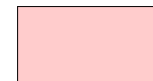
RFC 2689	RFC 3485	RFC 3544	RFC 3819	RFC 3940	RFC 3941	RFC 4629
RFC 4919	RFC 4944	RFC 5049	RFC 5401	RFC 5740	RFC 5856	RFC 5857
RFC 5858	RFC 6282	RFC 6469	RFC 6568	RFC 6606	RFC 6775	RFC 6690
RFC 7049	RFC 7228	RFC 7252	RFC 7388	RFC 7390	RFC 7400	RFC 7641
RFC 7668	RFC 7744	RFC 7925	RFC 7959	RFC 8075	RFC 8132	RFC 8152
RFC 8307	RFC 8323	RFC 8376	RFC 8392	RFC 8424	RFC 8516	...and more



Connectivity WGs

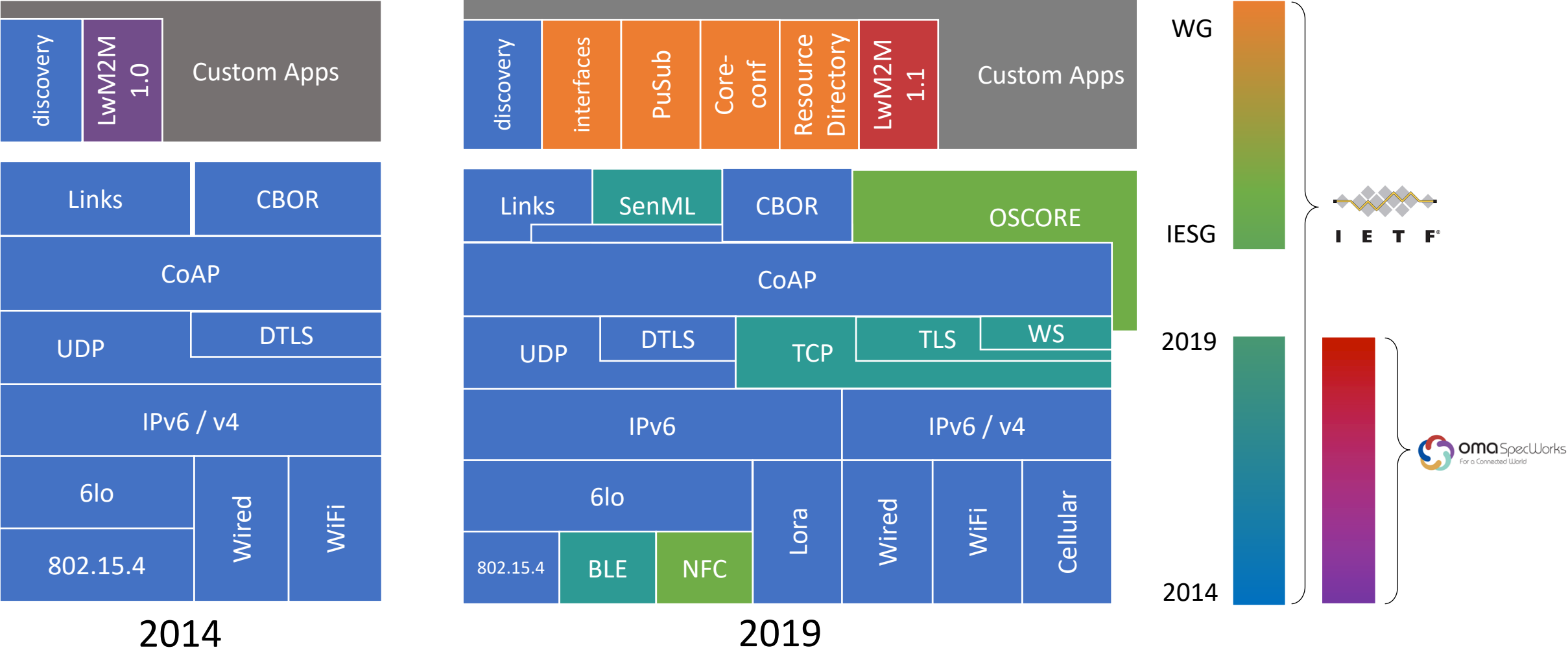


Application WGs



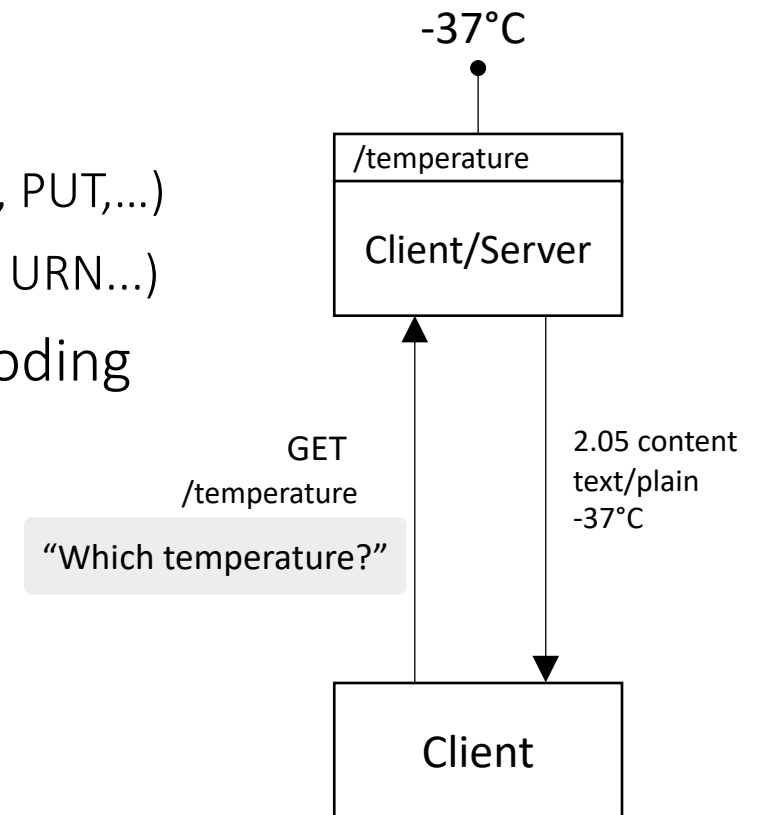
Security WGs

Standards Device Stack



The Constrained Application Protocol (CoAP)

- CoAP (RFC7252) implements HTTP's **REST** model
 - Simple devices: 100 to 250 KiB code and 10 to 50 KiB RAM
 - Each device can be client and server exposing resources
 - CoAP defines methods to access those resources (GET, POST, PUT,...)
 - Same key concepts borrowed from HTTP (Media types, URL, URN...)
- Has a compact 4-byte header, with simple options encoding
- Simple protocol, datagram (UDP, DTLS)
 - Reliability through header message type "*CON/NON*"
 - With TCP/TLS (RFC8323) support for NAT-ed environments
- The Resource Directory provides a directory service



The Constrained Application Protocol (CoAP)

- CoRELink (RFC6690) provides a link format

- Reuses Web Linking RFC5988 for IoT.
- Enables query parameters for discovery (lt, gt...)
- Enables attribute and relation types (rt, if, sz).

```
<3303/0/5700>;rt="oma:lwm2m:temp";ct="0"
```

- Notifications available through *observe* option (RFC7641)

- Can observe and add query parameters to the observation

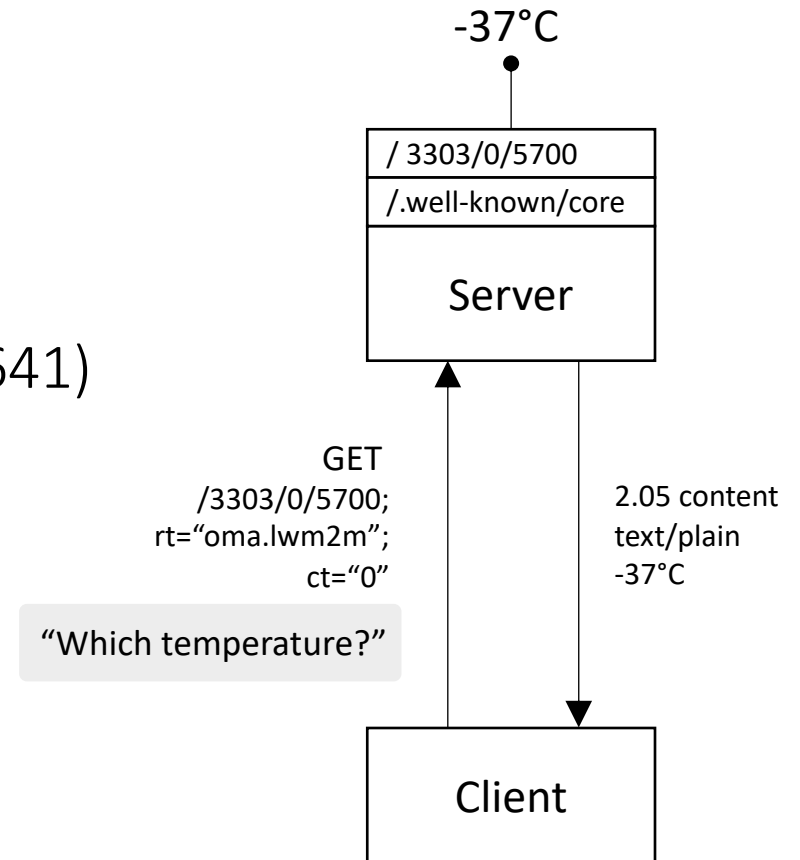
```
<3303/0/5700?lt=0>
```

- The *"/.well-known/core"* URI provides discovery

- Multiple serialization formats used with CoAP

- SenML (RFC8428): Minimalistic JSON
- CBOR (RFC7049): Binary serialization

- Multiple implementations available at coap.technology

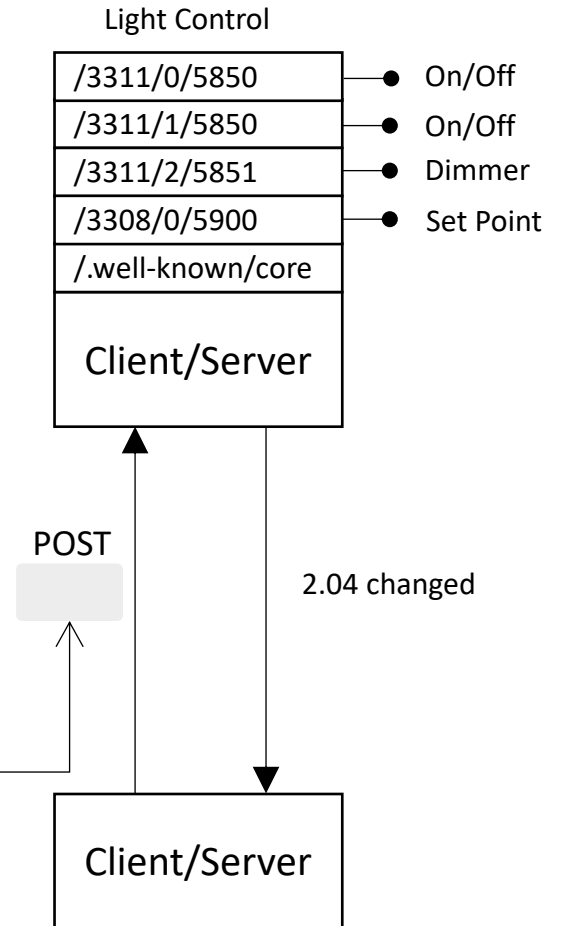


Example Actuation

- When using IPSO actuation is handled with executable (E) resources as well as readable and writeable ones (RW).
 - Parameters needed for actuation are passed along on resources.
 - Type and range of values are known to client based on the schema.
 - Actuation can use executable and writeable (EW) resources.

A server Write-Composite to switch off 2 light sources, dim a 3rd to 20% and set the thermostat to 18 degrees will have a JSON payload as shown in table below. Lights are all controlled by instances of IPSO Light Control Object (Object ID 3311), while thermostat is controlled by an instance of IPSO object Set Point (Object ID 3308).

```
[{"n": "/3311/0/5850", "vb": false},  
{"n": "/3311/1/5850", "vb": false},  
{"n": "/3311/2/5851", "v": 20},  
{"n": "/3308/0/5900", "v": 18}]
```



Serialization Formats

SenML - JSON

```
[{"bn":"/3/0/","n":"0","vs":"Open  
Mobile Alliance"},  
{"n":"1","vs":"Lightweight M2M  
Client"},  
{"n":"2","vs":"345000123"},  
{"n":"3","vs":"1.0"},  
{"n":"6/0","v":1},  
{"n":"6/1","v":5},  
{"n":"7/0","v":3800},  
{"n":"7/1","v":5000},  
{"n":"8/0","v":125},  
{"n":"8/1","v":900},  
{"n":"9","v":100},  
{"n":"10","v":15},  
{"n":"11/0","v":0},  
{"n":"13","v":1367491215},  
{"n":"14","vs":"+02:00"},  
{"n":"16","vs":"U"}]
```

SenML-CBOR

```
90 a3 21 65 2f 33 2f 30 2f 00 61 30  
03 74 4f 70 65 6e 20 4d 6f 62 69 6c  
65 20 41 6c 6c 69 61 6e 63 65 a2 00  
61 31 03 76 4c 69 67 68 74 77 65 69  
67 68 74 20 4d 32 4d 20 43 6c 69 65  
6e 74 a2 00 61 32 03 69 33 34 35 30  
30 30 31 32 33 a2 00 61 33 03 63 31  
2e 30 a2 00 63 36 2f 30 02 01 a2 00  
63 36 2f 31 02 05 a2 00 63 37 2f 30  
02 19 0e d8 a2 00 63 37 2f 31 02 19  
13 88 a2 00 63 38 2f 30 02 18 7d a2  
00 63 38 2f 31 02 19 03 84 a2 00 61  
39 02 18 64 a2 00 62 31 30 02 0f a2  
00 64 31 31 2f 30 02 00 a2 00 62 31  
33 02 1a 51 82 42 8f a2 00 62 31 34  
03 66 2b 30 32 3a 30 30 a2 00 62 31  
36 03 61 55
```

SenML-CBOR diagnostic

```
[{-2: "/3/0/", 0: "0", 3: "Open  
Mobile Alliance"}, {0: "1", 3:  
"Lightweight M2M Client"},  
{0: "2", 3: "345000123"},  
{0: "3", 3: "1.0"},  
{0: "6/0", 2: 1},  
{0: "6/1", 2: 5},  
{0: "7/0", 2: 3800},  
{0: "7/1", 2: 5000},  
{0: "8/0", 2: 125},  
{0: "8/1", 2: 900},  
{0: "9", 2: 100},  
{0: "10", 2: 15},  
{0: "11/0", 2: 0},  
{0: "13", 2: 1367491215},  
{0: "14", 3: "+02:00"},  
{0: "16", 3: "U"}]
```


CoAP Implementations

Overview

Specification

Implementations

Tools

CoAP

RFC 7252 Constrained Application Protocol

“The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the **Internet of Things**.

The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.”

coap.technology

coap-shell client

```
coap://coap.me:>quit  
✓ ~/Dev/projects/coap/coap-shell [master {origin/master}|+ 1]  
21:14 $ java -jar ./target/coap-shell-1.0.3-SNAPSHOT.jar  
  
      _--_    _--_    _--_    _--_    _--_  
 /  _/_/  _/  _/_/  _/  _/_/  _/  _/_/  _/_/  
/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/  
CoAP Shell (v1.0.3-SNAPSHOT)  
For assistance hit TAB or type "help".  
  
server-unknown:>connect coap://coap.me  
Connected to [coap://coap.me], [con], [non-observable]  
coap://coap.me:>ping  
available  
coap://coap.me:>discover
```

github.com/tzolov/coap-shell

<https://asciinema.org/a/wPnbU56v1R3nkMafnmzaiB20B>

go-coap server + coap-shell client

build **passing**  codecov 66% go report **A+**

CoAP Client and Server for go

Features supported:

- CoAP over UDP [RFC 7252](#).
- CoAP over TCP/TLS [RFC 8232](#)
- Observe resources in CoAP [RFC 7641](#)
- Block-wise transfers in CoAP [RFC 7959](#)
- request multiplexer
- multicast
- CoAP NoResponse option in CoAP [RFC 7967](#)

Not yet implemented:

- CoAP over DTLS

github.com/go-ocf/go-coap

<https://asciinema.org/a/pd13OzAlwVvos63HXqfGDTUPi>

Libcoap client and server

Version 4.2.0 is out! [Try now!](#)



C-Implementation of CoAP

libcoap implements a lightweight application-protocol for devices that are constrained their resources such as computing power, RF range, memory, bandwidth, or network packet sizes.

The Constrained Application Protocol (CoAP) was standardized in the [Internet Engineering Task Force \(IETF\)](#) as [RFC 7252](#).

[Learn more](#)

github.com/obgm/libcoap

<https://asciinema.org/a/I0yOK7e5qgMOQmXYmBUqp1Vbr>

All of them together:

<https://asciinema.org/a/AyO4fdJjakqTbVjhjFAnxbLQi>